

Docket No. AUS920000406US1

CLAIMS:

What is claimed is:

1. A method for editing text used in a user interface of a computer program, comprising the steps of:

5 receiving contextual information describing a visual context for the computer program's generation of the text;

 receiving the text from a first user interface text storage format;

10 combining the contextual information with the text to form a visual representation of a display in the user interface; and

 displaying the visual representation.

15 2. The method of claim 1, further comprising the step of:

 receiving editing instructions to edit the text; and

 based on the editing instructions, generating new text.

20 3. The method of claim 2, further comprising the step of:

 storing the new text in a second user interface text storage format.

Docket No. AUS920000406US1

4. The method of claim 1, wherein the first user interface text storage format is a catalog file.

5. The method of claim 1, wherein the first user interface text storage format is a resource bundle.

5 6. The method of claim 1, wherein the contextual information includes graphical user interface (GUI) rendering information.

7. The method of claim 1, wherein the computer program is written in Java.

10 8. A method for generating an editable representation of a user interface in an executing computer program, comprising the steps of:

detecting when the executing computer program generates a user interface feature; and

15 recording a description of the user interface feature.

9. The method of claim 8, further comprising the step of:

20 associating the description with an item of text in a user interface text storage format.

10. The method of claim 8, wherein the description includes information regarding the rendering of the user interface feature.

Docket No. AUS920000406US1

11. The method of claim 8, wherein the user interface is a graphical user interface (GUI).

12. The method of claim 8, wherein the user interface text storage format is a resource bundle.

5 13. The method of claim 8, wherein the executing computer program is written in Java.

14. A method for editing text used in a user interface of a computer program, comprising the steps of:

10 receiving a sequential record of system events that occurred during an execution session of the computer program;

15 executing an executable of the computer program, wherein the executable contains an introspective editor; and

reproducing the system events from the sequential record to control execution of the executable.

20 15. The method of claim 14, further comprising the step of:

responsive to a user input, suspending execution of the reproducing step.

Docket No. AUS920000406US1

16. The method of claim 15, further comprising the step of:

responsive to a second user input, resuming the execution of the reproducing step.

5 17. The method of claim 14, wherein the system events include at least one of a keystroke, a mouse click, a mouse double-click, and a mouse drag.

10 18. A computer program product, in a computer-readable medium, for editing text used in a user interface of a computer program, comprising instructions for:

receiving contextual information describing a visual context for the computer program's generation of the text;

15 receiving the text from a first user interface text storage format;

combining the contextual information with the text to form a visual representation of a display in the user interface; and

20 displaying the visual representation.

Docket No. AUS920000406US1

19. The computer program product of claim 18, further comprising instructions for:

receiving editing instructions to edit the text; and

5 based on the editing instructions, generating new text.

20. The computer program product of claim 19, further comprising instructions for:

10 storing the new text in a second user interface text storage format.

21. The computer program product of claim 18, wherein the first user interface text storage format is a catalog file.

15 22. The computer program product of claim 18, wherein the first user interface text storage format is a resource bundle.

23. The computer program product of claim 18, wherein the contextual information includes graphical user interface (GUI) rendering information.

20 24. The computer program product of claim 18, wherein the computer program is written in Java.

25. A computer program product, in a computer-readable medium, for generating an editable

Docket No. AUS920000406US1

representation of a user interface in an executing computer program, further comprising instructions for:

detecting when the executing computer program generates a user interface feature; and

- 5 recording a description of the user interface feature.

26. The computer program product of claim 25, further comprising instructions for:

- 10 associating the description with an item of text in a user interface text storage format.

27. The computer program product of claim 25, wherein the description includes information regarding the rendering of the user interface feature.

- 15 28. The computer program product of claim 25, wherein the user interface is a graphical user interface (GUI).

29. The computer program product of claim 25, wherein the user interface text storage format is a resource bundle.

- 20 30. The computer program product of claim 25, wherein the executing computer program is written in Java.

Docket No. AUS920000406US1

31. A computer program product, in a computer-readable medium, for editing text used in a user interface of a computer program, further comprising instructions for:

5 receiving a sequential record of system events that occurred during an execution session of the computer program;

 executing an executable of the computer program, wherein the executable contains an introspective editor; and
10

 reproducing the system events from the sequential record to control execution of the executable.

32. The computer program product of claim 31, further
15 comprising instructions for:

 responsive to a user input, suspending execution of the instructions for reproducing.

33. The computer program product of claim 32, further comprising instructions for:

20 responsive to a second user input, resuming the execution of the instructions for reproducing.

Docket No. AUS920000406US1

34. The computer program product of claim 31, wherein the system events include at least one of a keystroke, a mouse click, a mouse double-click, and a mouse drag.

5 35. A data processing system executing a program that displays text, the data processing system comprising:

a bus system;

a memory connected to the bus system, wherein the memory includes a set of instructions; and

10 a processing unit, wherein the processing unit includes at least one processor,

wherein the processing unit executes the set of instructions to perform the acts of:

15 receiving contextual information describing a visual context for the program's generation of the text;

receiving the text from a first user interface text storage format;

20 combining the contextual information with the text to form a visual representation of a display in the user interface; and

displaying the visual representation.

Docket No. AUS920000406US1

36. The data processing system of claim 35, wherein the processing unit executes the set of instructions to perform the additional acts of:

5 receiving editing instructions to edit the text; and

 based on the editing instructions, generating new text.

37. The data processing system of claim 36, wherein the data processing system executes the set of
10 instructions to perform the additional act of:

 storing the new text in a second user interface text storage format.

38. The data processing system of claim 35, wherein the first user interface text storage format is a
15 catalog file.

39. The data processing system of claim 35, wherein the first user interface text storage format is a resource bundle.

40. The data processing system of claim 35, wherein
20 the contextual information includes graphical user interface (GUI) rendering information.

41. The data processing system of claim 35, wherein the program is written in Java.

Docket No. AUS920000406US1

42. A data processing system to generate an editable representation of a user interface in an executing program, the data processing system comprising:

a bus system;

5 a memory connected to the bus system, wherein the memory includes a set of instructions; and

a processing unit, wherein the processing unit includes at least one processor,

10 wherein the processing unit executes the set of instructions to perform the acts of:

detecting when the executing program generates a user interface feature; and

recording a description of the user interface feature.

15 43. The data processing system of claim 42, wherein the data processing system executes the set of instructions to perform the additional act of:

associating the description with an item of text in a user interface text storage format.

20 44. The data processing system of claim 42, wherein the description includes information regarding the rendering of the user interface feature.

Docket No. AUS920000406US1

45. The data processing system of claim 42, wherein the user interface is a graphical user interface (GUI).

46. The data processing system of claim 42, wherein the user interface text storage format is a resource
5 bundle.

47. The data processing system of claim 42, wherein the executing program is written in Java.

48. A data processing system to edit text used in a user interface of a program, the data processing
10 system comprising:

a bus system;

a memory connected to the bus system, wherein the memory includes a set of instructions; and

a processing unit, wherein the processing
15 unit includes at least one processor,

wherein the processing unit executes the set of instructions to perform the acts of:

receiving a sequential record of system events that occurred during an execution
20 session of the program;

executing an executable of the program, wherein the executable contains an introspective editor; and

Docket No. AUS920000406US1

reproducing the system events from the sequential record to control execution of the executable.

49. The data processing system of claim 48, wherein
5 the data processing system executes the set of instructions to perform the additional act of:

responsive to a user input, suspending execution of the instructions for reproducing.

50. The data processing system of claim 49, wherein
10 the data processing system executes the set of instructions to perform the additional act of:

responsive to a second user input, resuming the execution of the instructions for reproducing.

- 15 51. The data processing system of claim 48, wherein the system events include at least one of a keystroke, a mouse click, a mouse double-click, and a mouse drag.